CORPUS

Corpus

6 | 2007
Interprétation, contextes, codage

CorpusReader : un dispositif de codage pour articuler une pluralité d'interprétations

Sylvain Loiseau

OpenEdition Journals

Édition électronique
URL : http://journals.openedition.org/corpus/1282
ISSN : 1765-3126

Éditeur
Bases ; corpus et langage - UMR 6039

Édition imprimée
Date de publication : 1 décembre 2007
Pagination : 153-186
ISSN : 1638-9808

Référence électronique
Sylvain Loiseau, « CorpusReader : un dispositif de codage pour articuler une pluralité d'interprétations », *Corpus* [En ligne], 6 | 2007, mis en ligne le 02 juillet 2008, consulté le 20 avril 2019.
URL : http://journals.openedition.org/corpus/1282

CorpusReader : un dispositif de codage pour articuler une pluralité d'interprétations

Sylvain LOISEAU
LIMSI/CNRS

1. Introduction[1]

Cet article présente un dispositif expérimental[2], CorpusReader[3], élaboré pour construire des corpus articulant plusieurs niveaux de description (morphologique, morphosyntaxique, syntaxique, etc), appelés ici « multi-annotés », et pour créer des observables articulant ces niveaux de description.

L'hypothèse à laquelle répond ce dispositif est que des corrélations entre niveaux de description peuvent caractériser une norme linguistique (comme un genre, un discours ou un idiolecte[4]) plus finement que chacun des niveaux pris isolément. L'enjeu des observables ainsi produits est donc de pouvoir rapporter des régularités à des normes linguistiques. Or, si l'annotation automatique de nombreux niveaux de description est devenu un acquis, et si les analyseurs, notamment morpho-syntaxiques, se stabilisent en instruments (Habert, 2005a), la simple mise en regard de plusieurs annotations d'un même texte est encore difficile à réaliser. Pourtant, des progrès significatifs à court terme de ces instruments sont sans doute peu probables, au moins au niveau morphosyntaxique, tandis que des corpus

1. J'ai plaisir à remercier Marie Guegan, Benoît Habert, Serge Heiden et Bénédicte Pincemin pour leurs relectures.

2. Habert (2005a).

3. Ce dispositif a été développé comme méthodologie d'une thèse (Loiseau, 2006), et a été utilisé dans plusieurs travaux, notamment Loiseau *et al.* (2006), Poudat (2006) ; il est enfin utilisé dans le cadre d'un projet RNRT (projet Autograph). Une documentation complète ainsi que de nombreux exemples et le code source sont disponibles en ligne : <http://panini.u-paris10.fr/~sloiseau/CR>. L'ensemble du logiciel est sous licence BSD (logiciel « libre »).

4. Voir Coseriu (2001), Rastier (2005), et ci-dessous.

associant les annotations de ces instruments permettraient d'accéder, par l'observation des corrélations entre niveaux, à des régularités essentielles et peu décrites.

L'objectif de CorpusReader est donc de permettre la réalisation matérielle et l'exploitation d'une multi-annotation.

Cette mise en relation d'annotations consiste, dans une certaine mesure, à créer de la contextualité. Par opposition aux corpus mono-annotés, où le contexte est réduit à une dimension syntagmatique (le voisinage d'une forme) et à un même niveau de description, un corpus multi-annoté contextualise une unité par d'autres niveaux de description. D'autre part, il articule des paliers différents, c'est-à-dire des contextes de tailles variables, du morphème aux divisions du texte.

Dans cette perspective, les différents niveaux offrent chacun des vues limitées ou partielles sur le texte. La division en paliers et en niveaux de description est une division méthodologique, héritée de la spécialisation des analyseurs dans des domaines empiriques suffisamment restreints pour être instrumentalisables ; cette division n'est pas pour nous une hypothèse sur la structuration de l'objet linguistique lui-même. Les annotations ont donc elles-mêmes un statut d'interprétation.

En résumé, le *contexte* est entendu ici comme l'articulation, dans un dispositif de codage, d'une pluralité d'annotations qui réifient des interprétations.

Puisque les annotations sont autant d'interprétations, le dispositif doit notamment permettre d'annoter un même niveau par plusieurs analyseurs « concurrents » sur ce niveau. Cette mise en parallèle de plusieurs analyses permet d'associer des sous-ensembles de chacune des annotations. Par exemple, l'utilisation conjointe des analyseurs Cordial et Syntex permet d'utiliser, dans une requête, les étiquettes morphosyntaxiques du premier et les unités syntagmatiques du second, issues d'une analyse robuste en dépendances[5]. Des phénomènes irréductibles à chacune des annotations prise isolément peuvent ainsi être construits.

5. C'est plus généralement une complémentarité entre paradigmes qui peut être envisagée : notamment entre une analyse fine mais peu adaptable et une analyse impliquant un apprentissage endogène.

Plusieurs architectures pour l'annotation de corpus ont été proposées[6]. Aucune architecture ni même aucune méthodologie ne fait cependant encore consensus et les différences entre les choix de codage impliquent de larges divergences non seulement quant aux capacités descriptives mais également quant à la nature de l'objet empirique constitué. Par rapport aux architectures existantes, les spécificités de CorpusReader découlent principalement de l'importance accordée au format. L'objectif de CorpusReader a été de ne pas limiter, autant que possible, la diversité des phénomènes représentables, observables et quantifiables. Il est donc spécialisé dans la simple manipulation d'un format[7] afin de donner à l'interprète la complète maîtrise des objets construits. Cette construction a deux versants : d'une part, fusionner et faire coexister plusieurs annotations sans limitation sur la complexité des données ; d'autre part, réaliser des requêtes portant sur toutes les propriétés des données.

CorpusReader n'a donc aucune « compétence » linguistique ni statistique : sa fonction est de se situer entre deux types d'outils disponibles par ailleurs, analyseurs linguistiques et logiciels statistiques, pour permettre de faire accéder à des traitements quantitatifs des corpus multi-annotés.

Après avoir précisé puis illustré certaines de ces notions sur un exemple simplifié, cet article présentera les deux versants de CorpusReader : construction d'un corpus multi-annoté, puis expression de requêtes sur le corpus construit.

2. Normes, quantification et corpus multi-annoté

Nous désignons par norme, de façon très générale, les propriétés d'un texte qui ne relèvent pas du système fonctionnel de la langue, et ne peuvent donc être décrites en termes

6. On peut citer par exemple GATE (Bontcheva *et al.*, 2004), NITE (Carletta *et al.*, 2003), Atlas (Bird *et al.*, 2000) et LinguaStream (Bilhaut & Widlöcher, 2006).

7. Plus précisément, CorpusReader manipule non pas le *format* XML, mais le *modèle de données* abstrait d'XML, à savoir l'Infoset XML, décrit dans Cowan & Tobin (2004).

d'oppositions distinctives. Norme s'entend ici au sens de normal, pas de normatif[8]. S'articule ainsi une pluralité de normes, interne à la langue :

> Une langue historique n'est jamais un seul « système linguistique » mais un « diasystème » : un ensemble de « systèmes linguistiques », entre lesquels il y a à chaque pas coexistence et interférence. (Coseriu, 2001 : 240).

Les descriptions reposant sur de grands corpus textuels ont montré également la pertinence de la description de la variation des normes d'un point de vue quantitatif[9]. La description de normes est maintenant un objectif largement partagé au sein des linguistiques de corpus (pour un exposé des enjeux, on peut se rapporter à Habert & Zweigenbaum, 2002).

Par rapport à d'autres termes (type de texte, registre, régularité, sous-langage, etc.) le terme de norme permet d'éviter un antagonisme entre règle (formalisme) et fréquence (régularité) et de problématiser la relation entre fait quantitatif et objet linguistique (« qualitatif »)[10]. Dans une acception plus précise, à la suite de Rastier (2005), on parlera de norme au sens d'une instance praxéologique (c'est-à-dire instaurée par des pratiques, des usages) donnant lieu à une variation : un texte

8. « [...] la *norme*, c'est-à-dire [...] la réalisation normale [du système] dans la communauté linguistique [...] » (Coseriu, 2001 : 110). L'opposition entre norme prescriptive et norme descriptive n'est cependant qu'une question de point de vue : toute norme objective tend à « persévérer dans son être » et génère des prescriptions (Glessgen, 2007 : 95).

9. La relation entre norme et description quantitative est soulignée déjà par Coseriu (Lara, 1983 : 19).

10. Ce que formule ainsi Canguilhem : « [le terme normal] désigne tantôt un fait capable de description par recensement statistique - moyenne des mesures opérées sur un caractère présenté par une espèce [...] - et tantôt un idéal, principe positif d'appréciation, au sens de prototype ou de forme parfaite » (2003 :200), ou, dans le domaine linguistique, Guiraud : « [...] nous rencontrons deux problèmes selon que nous considérons la norme en termes quantitatifs, comme l'emploi le plus fréquent ; ou en termes qualitatifs, comme l'emploi le plus conforme à la structure du système. [...] En fait, norme quantitative et norme qualitative, tendent à se confondre dans la grande majorité des cas. » (1969 : 61).

relève de normes sociolectales (genre et discours) et d'une norme idiolectale (style individuel). Chaque norme est alors comme une dimension d'un « espace des normes » dans lequel chaque texte se situe.

Au sein du « variationnisme renouvelé » (Habert & Zweigenbaum, 2002 : 98) que permettent les linguistiques de corpus, la relation entre norme et description quantitative se pose en effet à deux niveaux et dessine un cercle vertueux. D'une part les méthodes quantitatives sont indispensables à la description des normes, puisqu'elles ne peuvent être décrites en termes d'oppositions fonctionnelles. D'autre part la prise en compte des normes est un préalable à toute description quantitative, puisque la norme est précisément l'unité qui permet de donner un sens à une fréquence[11].

De plus, la question des normes remet également en question une description en termes de niveaux et de paliers autonomes (Rastier, 2005). La distinction de niveaux de description est le résultat d'une analyse en termes de système fonctionnel : non seulement cette distinction n'est pas nécessaire dans le cadre de la description d'une norme, mais elle peut même y faire obstacle en réduisant drastiquement la combinatoire des régularités observables.
C'est ce que souligne par exemple F. Gadet :

> Cette perspective, qui oblige à prendre en compte les énoncés selon des principes de différentes ordres, devrait renouveler la définition des genres en les montrant comme des faisceaux de

11. Par exemple : « [...] la statistique linguistique porte à des résultats linguistiquement inutilisables et même statistiquement faux, du fait qu'elle considère souvent toute une langue historique ou toute une langue commune comme un seul « continuum ». Ainsi, il n'y a pas de sens à établir la fréquence relative de *may* par rapport à *can* dans la langue anglaise toute entière, si l'on constate que *may* « peut varier jusqu'à zéro ». En réalité, cela signifie qu'il y a au moins deux types d'anglais à distinguer : l'un dans lequel l'opposition *can* - *may* existe [...], et un autre type, dans lequel seul *can* se présente et dans lequel il est absurde de constater une « proportion » entre les deux termes de l'opposition, puisque l'opposition même n'existe pas. » (Coseriu, 2001 : 243, note).

> paramètres, et non plus des rubriques rhétoriques ou situationnelles héritées de la tradition. (2003 : 59)

ou, dans la perspective du traitement automatique des langues, Habert & Zweigenbaum :

> [Le traitement automatique effectif des langues] enjoint aussi de munir les données attestées d'annotations fines, multiples, permettant de progresser vers les régularités sous-jacentes. (2002 : 99).

CorpusReader est donc un dispositif pour faciliter l'exploration de régularités et la description de normes dans des corpus multi-niveaux.

3. Un exemple

Un exemple simplifié permettra d'illustrer le type d'observables que ce dispositif tente de rendre possible.

Dans cet exemple il s'agit d'étudier la variation diachronique dans un corpus constitué de quinze textes de Gilles Deleuze[12]. C'est principalement la variation du système énonciatif qui a été observée. Pour cela, chaque texte a été caractérisé par plusieurs variables, relevant de plusieurs niveaux de description ou de plusieurs sémiotiques : fréquences relatives des temps et des modes, des pronoms personnels, des signes de ponctuation, des noms propres parmi l'ensemble des noms, enfin de différentes marques de segmentation textuelle (italiques, citations, divisions, titres, paragraphes). Certaines variables représentent des informations encore plus « fines », comme les proportions relatives de traits « indéfinis » et « définis » dans les déterminants.

Une analyse en composantes principales (ACP) a permis de caractériser les textes par des ensembles de variables.

12. Ce corpus a fait l'objet de la thèse citée (Loiseau, 2007).

Sur le plan issu des deux premiers facteurs (ci-dessous) les textes sont reliés par des flèches dans l'ordre chronologique. Ils s'ordonnent chronologiquement le long du premier facteur[13].

Du point de vue des variables, tous les niveaux de description représentés varient sur les deux axes représentés. Par exemple, du point de vue des personnes, les textes sont caractérisés successivement par la première personne du pluriel (le *nous* académique), puis les secondes personnes, enfin par le *je* et le *on* dans les derniers textes. Sur le plan des signes de ponctuation, les premiers textes sont caractérisés par le point virgule, les textes centraux par le point d'exclamation, enfin les derniers textes par les points de suspension. Les marques de structuration du texte comme les italiques (*hi*), titres (*head*), ou notes sont plus fréquentes dans les premiers textes. Les temps et modes les plus caractéristiques sont d'abord le subjonctif présent (*subp*), puis l'impératif (*imp*), et enfin le conditionnel (*cond*).

On observe donc des corrélations dans la variation des différents niveaux sur les deux axes représentés : l'ensemble des niveaux contribue à opposer le discours académique des premiers textes, le discours politisé des textes centraux, et le discours littéraire des derniers textes.

13. Les exceptions apparentes sont dues à des alternances entre essais et commentaires : entre deux textes aux dates de publication très proches, le commentaire paraît plus « conservateur » et recule donc sur l'axe diachronique.

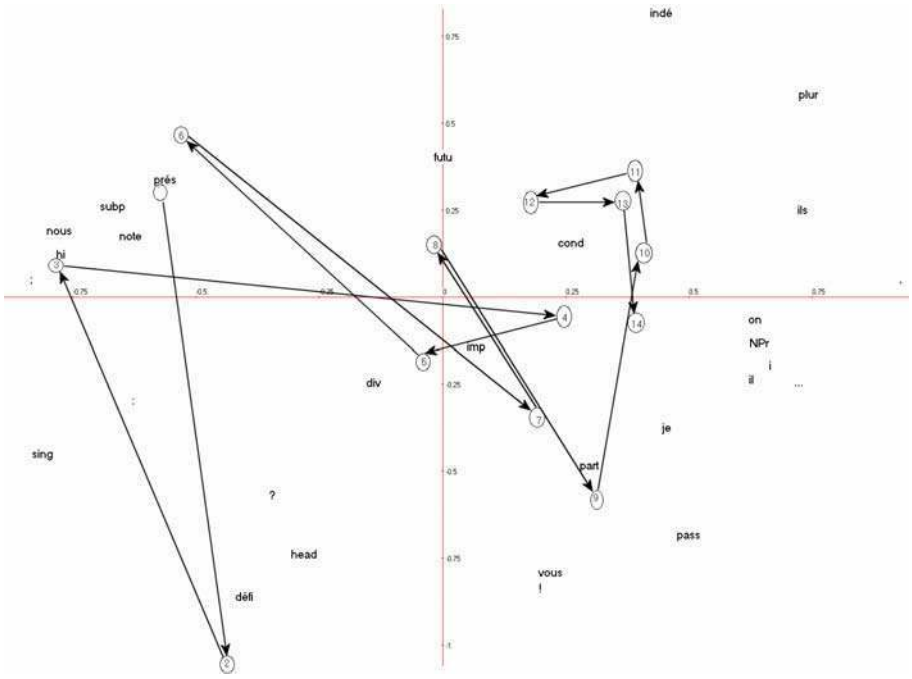

Figure 1 : ACP du corpus G. Deleuze.

Dans cet exemple, à la portée descriptive nécessairement limitée, l'analyse porte sur le palier du texte et mobilise plusieurs niveaux : morphologique, lexical, sémantique (opposition de l'indéfini et du défini), ponctèmes, qualités typographiques et structurations. L'ensemble des phénomènes quantifiés est ici issu d'un seul analyseur (Cordial), à l'exception des variables portant sur la structure du texte : une « fusion » de ces informations est nécessaire. Enfin, la requête produit ici une matrice (une table de contingence) ; d'autres structures de données peuvent être construites en sortie, comme des graphes ou des listes de fréquence.

Pour décrire des corrélations entre les différents niveaux de description, il est donc nécessaire de représenter ensemble, au moyen d'une structure de données suffisamment expressive, des annotations portant sur plusieurs niveaux. Il est également nécessaire de pouvoir exprimer des requêtes pour

filtrer, désigner, décompter, et recomposer à façon de nouvelles structures de données.

4. Enjeu du format de représentation

Le format de représentation du corpus doit répondre à deux exigences principales : une exigence d'expressivité, de façon à permettre la coexistence d'une pluralité d'annotations, et une exigence de standardisation, de façon à bénéficier des outils existants et garantir une pérennité. Ce cahier des charges a conduit à adopter le format XML et le vocabulaire de la *Text Encoding Initiative*.

4.1. XML comme format d'accumulation

Les bénéfices essentiels d'XML pour la constitution d'un corpus multi-niveaux tiennent à sa qualité de format « d'intégration ». On peut distinguer différents types d'intégrations permises par XML :

- C'est un format permettant de représenter à la fois des documents dits « orientés données » (*data oriented*) et des documents « orientés document » (*narrative oriented*)[14]. Il permet donc de joindre dans un corpus des représentations « naïves » ou simples (par exemple une structuration en paragraphes) et des annotations à différents niveaux de structuration, depuis des lemmatisations associées à des segmentations jusqu'à des structures de traits associées à des analyses morphosyntaxiques. Un corpus peut donc associer des annotations relevant de différents niveaux de formalisation ; cela permet d'une part une formalisation progressive, et d'autre part de croiser, dans des requêtes, différents niveaux de formalisation.

- La distinction entre « données » (par exemple, les chaînes de caractères initiales) et « méta-données » (les

14. Un document « orienté données » est structuré et régulier, et pourrait être représenté également par une base de données. Un document « orienté document » a une structure plus « naïve », moins simple et régulière. Voir Sperberg-McQueen (2005). On peut dire également que dans un document « orienté document », contrairement à un document « orienté données », il y a un ordre donné des nœuds textes : c'est l'axe de la linéarité du signifiant.

annotations) est révocable : les « méta-données » peuvent être utilisées elles-mêmes comme des variables pour des explorations quantitatives. Le modèle de données d'XML[15] rend accessible à l'observation toutes les propriétés du corpus, y compris les catégories de l'analyse et les choix interprétatifs. Toutes les propriétés de la structure de l'arbre, et donc la modélisation effectuée elle-même, sont accessibles via un langage standard comme XPath. *A contrario*, dans le cas d'une base de données relationnelle (SGBDR), la modélisation de l'objet induite par la structure de la base de données n'est pas accessible directement[16].

- L'annotation d'un document XML peut être étendue à différents moments dans le temps : la structure arborescente impliquée est plus flexible que, par exemple, celle d'une base de données relationnelle. Ainsi de nouvelles annotations peuvent réifier des interprétations fondées sur l'annotation existante dans un processus cumulatif.

- Un format arborescent exprime la précédence, c'est-à-dire, ici, la linéarité du signifiant (contrairement à une base de données, où la linéarité ne peut être exprimée qu'indirectement). Il est ainsi possible de conserver la dimension syntagmatique du contexte[17].

Les avantages d'XML tiennent aussi à sa diffusion, qui implique la disponibilité de nombreux outils pour manipuler et traiter ce format. La diffusion du format a également un enjeu plus scientifique en garantissant une pérennité des données, en facilitant leur échange, et donc en donnant une dimension cumulative aux descriptions de corpus.

15. L'« Infoset »XML (Cowan & Tobin, 2004).

16. Ceci pour des raisons de performance. On peut naturellement accéder aux propriétés de la base de données via les informations d'administration, ou en interrogeant, dans une seconde passe, le résultat d'une requête, mais c'est une information indirecte ou de « second rang », qui ne peut être mêlée à une requête sur les objets de la base eux-mêmes.

17. Voir Loiseau (2006).

4.2. Le vocabulaire de la Text encoding initiative

XML est en lui-même un formalisme arborescent et doit être complété par un vocabulaire, c'est-à-dire une convention sur la structure de l'arbre et les noms de ses nœuds. Une convention de représentation des données linguistiques dans le cadre d'XML est un travail coûteux et nécessairement collaboratif, du fait d'une part de la complexité de la tâche et d'autre part de la nécessité d'intégrer, dans l'élaboration d'une telle convention, une exigence de standardisation qui permette la pérennité et l'échange des données. Cette standardisation a donc à la fois des implications méthodologiques (réutiliser des outils et des langages existants) et des implications scientifiques (adopter un sociolecte transdisciplinaire dans la représentation des données, de façon à permettre l'échange des données et la cumulativité des résultats). Par ailleurs, une convention de représentation est une interprétation, quand bien même elle est érigée en standard : elle comporte nécessairement des choix définitoires sur la nature des données qu'il faut expliciter pour pouvoir au besoin s'en démarquer.

Pour l'annotation de corpus, le vocabulaire de la *Text Encoding Initiative* (TEI) est un bon candidat :

- c'est un travail collaboratif regroupant linguistes, éditeurs et bibliothèques, sous l'égide d'un « Consortium ». Elle inclut une exigence de standardisation : l'objectif de ce Consortium est d'élaborer un vocabulaire XML pour la communauté des sciences humaines. La TEI est intimement liée à XML (Sperberg-McQueen est un éditeur des deux recommandations, cf. Bray *et al.*, 2000 et Sperberg-McQueen & Burnard, 2001). Certains sous-ensembles de la TEI ont fait l'objet d'une standardisation : par exemple, le mécanisme des pointeurs (XPointer) a été adopté par le W3C[18] ; le mécanisme proposé pour l'annotation des structures de traits est en voie de normalisation (ISO).
- La TEI propose des conventions d'annotation pour l'ensemble des niveaux linguistiques (ainsi que pour d'autres sémiotiques :

18. World Wide Web, organisme de standardisation.

propriétés typographiques, poétiques, etc.)[19] C'est donc un vocabulaire adapté à une annotation multi-niveaux.
- Les *Recommandations*[20] de la TEI s'accompagnent d'une large documentation, d'une organisation et d'une classification des éléments recensés, et de nombreux outils permettant leur manipulation et leur exploitation.
- Elle propose également des mécanismes d'extension et de personnalisation, pour les besoins non couverts.

Enfin, ce qui est peut être plus important encore, la TEI inclut une véritable dimension philologique en proposant de définir un critère de qualité d'une annotation non pas seulement en termes de formalisme (de conformité du corpus annoté à une structure et à un vocabulaire définis formellement dans le cadre d'XML, par exemple une DTD ou un schéma), mais en termes d'explicitation, dans l'en-tête d'un corpus, des choix théoriques et pratiques qui ont présidé à sa constitution :

« Les *Recommandations* fournissent des moyens pour documenter l'annotation de telle sorte que le lecteur d'un texte puisse connaître les raisonnements qui ont présidé à cet encodage [*the reasoning behind that encoding*], et les choix interprétatifs généraux [*general interpretive decisions*] sur lesquels il est basé. »[21]

Un sous-ensemble important du vocabulaire est ainsi prévu pour associer à un texte une documentation des choix d'annotation. En d'autres termes, le formalisme d'XML est

19. Ce vocabulaire est à la fois une recension aussi exhaustive que possible des phénomènes susceptibles d'être annotés, et une proposition de représentation en XML pour chacun de ces phénomènes : « L'objectif de la TEI est de se saisir de deux questions : *quelles* sont les propriétés d'un texte [*textual features*] qui peuvent [*should*] être annotées (c'est-à-dire rendues explicites) dans une édition électronique [*electronic text*], et *comment* ces propriétés doivent-elles être représentées de façon à permettre des échanges sans dégradation [*loss-free*], et indépendants des environnements techniques [*platform-independent*] » (Burnard, 1995, italiques de l'auteur).

20. La première édition des *Recommandations* a été publiée en 1994. L'édition actuelle est la quatrième, publiée en 2002 ; une cinquième édition est en préparation.

21. <http://www.tei-c.org/P4X/AB.html>.

entendu comme un outil pour définir un critère de qualité de l'annotation et non comme un critère de qualité à lui seul : la qualité d'un annotation TEI est rapportée en dernière instance à la conformité de l'annotation non pas avec un schéma XML, mais avec les objectifs et les interprétations explicités de l'annotateur. Ce point est essentiel pour la définition de pratiques philologiques outillées – mais également pour assurer une véritable pérennité des corpus.

Cependant, CorpusReader s'écarte de la TEI sur deux points.

Premièrement, l'annotation, selon la TEI, n'est pas fondamentalement envisagée comme une interprétation, ni même une description, mais plutôt comme une désambiguïsation de faits déjà donnés : l'annotation consiste en « […] des marques explicites [*explicit markers*] des propriétés implicites des textes [*implicit textual features*] »[22]. L'enjeu est donc de rendre accessible (non ambigu) pour une machine ce qui serait présent de façon ambiguë dans les textes. Or, on peut également considérer une annotation comme constituant déjà

22. Voir <http://www.tei-c.org/P4X/AB.html> ; la notion d'interprétation n'intervient que dans une distinction entre ce qui serait une « représentation » et ce qui serait une « interprétation », distinguées sur le critère social du consensus, et cette distinction est exclue du périmètre de la TEI pour des raisons pratiques. « Dans ces *Recommandations*, aucune distinction forte et définitive n'est faite entre des données qui seraient `subjectives´ et des données qui seraient `objectives´, ou entre la `représentation´ et l'`interprétation´. Ces distinctions, bien que courantes et utiles dans des contextes étroits et bien définis, sont peut-être plutôt à considérer comme une distinction entre ce qui fait l'objet d'un consensus académique [*issues on which there is a scholarly consensus*] et ce qui ne fait pas l'objet d'un tel consensus. Ces consensus ont été et seront, à n'en pas douter, remis en cause [*subject to change*]. Les *Recommandations TEI* ne font aucune suggestion ni restriction sur les traits qui doivent être encodés. Les termes `descriptif´ et `interprétatif´, appliqués à différents types d'annotation dans les *Recommandations* n'impliquent aucune position particulière sur cette question théorique, mais reflète une distinction purement pratique entre les deux tâches que sont la tâche de la représentation du texte [*Committee on Text Representation*] et la tâche de l'interprétation et de l'analyse du texte [*Committee on Text Interpretation and Analysis*]. » (<http://www.tei-c.org/P4X/AB.html#ABDPIU>).

une interprétation ; en effet elle est réalisée en fonction des attentes descriptives et donc relativement à une précompréhension des textes.

En second lieu, la TEI propose une définition du texte qui peut sembler trop relative au formalisme arborescent. Dans le cadre de la TEI, la structure de données d'XML est utilisée non seulement comme un « format », mais aussi comme « modèle » théorique, c'est-à-dire que les propriétés formelles de cette structure de données reçoivent un statut et une justification théorique, et non seulement méthodologique.

> On peut sans doute dessiner de nombreux arbres de ce type [cf. figure ci-dessous] pour décrire la structure de cette anthologie. Certains de ces arbres peuvent être représentés comme une division supplémentaire dans l'arbre : par exemple, on peut diviser une ligne en mots, puisque aucun mot ne franchit la frontière du vers. C'est peut être étonnant, mais cette vue grossière et simplifiée de ce qu'est un texte (auquel Renear *et al.* ont donné le nom de « Hiérarchie ordonnée d'unités de contenu » [*ordered hierarchy of content objects*] (OHCO)) se révèle être très adaptée à la majorité des situations.[23]

23. Sperberg-McQueen & Burnard (2001) ; voir aussi <http://www.tei-c.org/P4X/SG.html>.

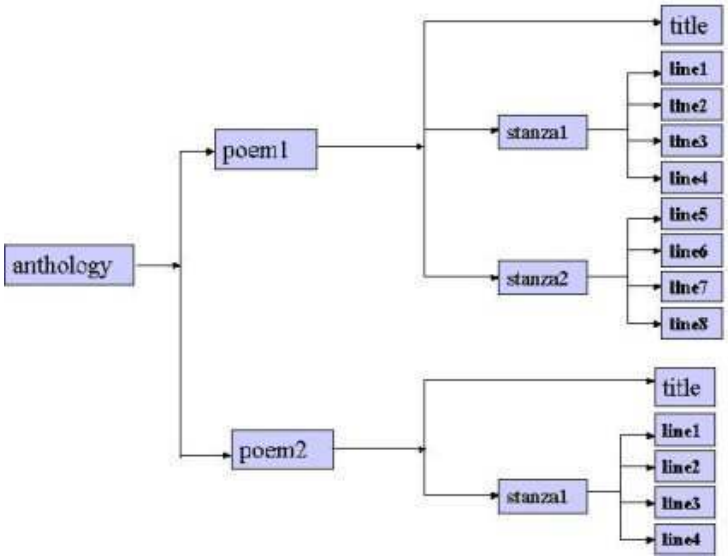

Figure 2 : représentation d'une anthologie de poèmes dans les *Recommandations TEI* (version 4).

Or, si XML code les données sous forme d'arbre, le format de données arborescent n'implique pas que les données soient elles-mêmes considérées comme hiérarchiques. Par exemple, on peut très bien modéliser une table, ou une base de données relationnelle, qui ne sont pas des structures hiérarchiques, au moyen d'un format de données hiérarchique. XML peut être utile parce que le format de modélisation en arbre est expressif, sans pour autant que les données soient elles-mêmes hiérarchiques.

5. Contextualiser avec un codage « embarqué »

XML est donc utilisé par CorpusReader comme un format, sans faire de la nature arborescente de la structure de données un modèle théorique.

Une interprétation forte du modèle arborescent est d'autant moins utile que, en pratique, de nombreux problèmes d'« enchevêtrement de hiérarchies » (*overlapping hierarchies* ou *overlapping structures*) perturbent la structure hiérarchique. Par exemple, les pages et les paragraphes ne sont pas

enchâssées et ne peuvent donc être notés comme des nœuds du même arbre ; il en va de même pour la segmentation en vers et en phrases dans les deux derniers vers du *Dormeur du val*[24] :

> *Il dort dans le soleil, la main sur sa poitrine*
> *Tranquille. Il a deux trous rouges au côté droit.*

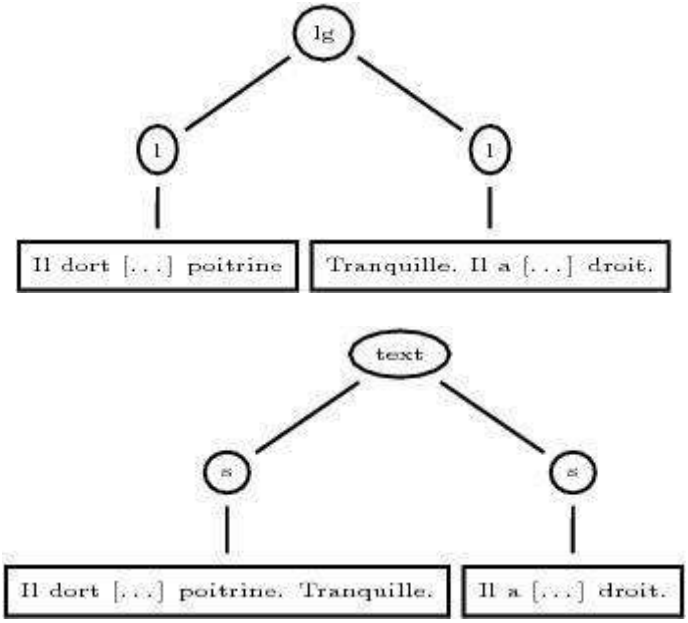

Figure 3 : deux segmentations concurrentes des deux derniers vers du *Dormeur du val*. L'élément « lg » (*line group*) regroupe les vers (« l », *line*) du tercet. Les phrases sont notées avec l'élément « s ».

Dans le cadre d'un corpus multi-niveaux, ces problèmes d'enchevêtrement ne sont plus cantonnés à quelques cas (comme l'incompatibilité entre les segmentations en pages et en paragraphes), mais deviennent omniprésents. Les analyses concurrentes d'un même niveau multiplient également les décalages (les seules segmentations en mots de deux analyseurs

24. Exemple proposé par Habert (2005b).

génèrent par exemple de nombreux enchevêtrements[25]). Enfin, même dans les cas où une préservation de la hiérarchie serait formellement possible, elle est lourde à obtenir automatiquement lors de la fusion de nombreuses annotations.

Une solution robuste à cette difficulté est donc nécessaire. Les différentes solutions[26] se ramènent à deux approches principales : d'un côté l'annotation dite « débarquée », de l'autre l'annotation en « nœuds-bornes » (balises *milestone*). La question du contexte souligne les enjeux de ce choix de codage : selon la solution adoptée, la contextualité créée est différente.

Dans le premier cas, on crée autant de documents différents (d'arborescences indépendantes) que de systèmes d'annotations potentiellement incompatibles. Par exemple, les analyses de différents analyseurs seront notées dans des documents distincts. Chaque document ne duplique pas les données communes (définie souvent comme le texte original), mais contient seulement les annotations propres à l'analyseur ; il pointe sur les données communes au moyen d'un système de références :

> L'annotation est dite débarquée [*stand-off*], ou externalisée [*external*], quand les éléments sont placés à l'extérieur du texte annoté : à un autre endroit du même fichier, ou même dans un autre fichier. L'annotation `pointe´ sur, plutôt qu'elle encadre, le contenu pertinent (*Recommandations TEI* (P 5), chapitre 14).

Dans le second cas, on remplace l'un des deux éléments XML en conflit par une paire de *nœuds-bornes*, c'est-à-dire deux éléments sans contenu qui notent le début et la fin de la portion de texte délimitée (voir figure 4). L'unité est donc désignée par

25. Le palier du mot n'est pas plus consensuel qu'un autre : la campagne GRACE d'évaluation d'analyseurs morpho-syntaxique du français a montré qu'un même corpus pouvait être segmenté en 411 750 mots par un analyseur et 463 600 par un autre (Habert, 2005b).

26. Voir le chapitre 31 des *Recommandations TEI* pour une recension, ainsi que DeRose (2004) et Barnard *et al.* (1992).

des bornes sur l'axe de la précédence, et non plus par un ancêtre sur l'axe de la dominance[27].

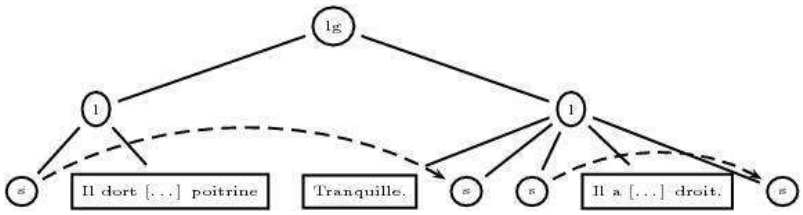

Figure 4 : une notation « en bornes » d'une segmentation en phrases dans le *Dormeur du val*.

Ces deux solutions s'opposent : la première préserve l'expression de la dominance au prix d'une perte de l'intégration des annotations, et donc d'une perte de contextualité. L'interrogation de corrélations entre deux annotations débarquées doit être prévue dès la conception de l'annotation, puisqu'elle est limitée aux points où les deux annotations sont alignées[28] ; les hiérarchies peuvent toujours être « réalignées », mais il faut alors en passer par des mécanismes externes à XML, c'est-à-dire faire reposer la contextualisation sur des dispositifs étrangers à la structure de données et techniquement coûteux à mettre en œuvre. De plus, ce dispositif suppose une distinction forte entre données primaires (qui sont communes aux différents documents) et données secondaires (qui sont spécifiques), distinction dont l'abandon est pourtant un avantage d'XML. Enfin une annotation débarquée est envisageable uniquement si le nombre

27. Voir DeRose (2004) et Barnard *et al.* (1992) pour une recension des variétés de nœuds-bornes.

28. Teich *et al.* (2001) notent la segmentation en phrases, la segmentation en mots, la segmentation en groupes de souffles, par autant de documents distincts. Les corrélations explorées ont dû être envisagées *a priori* pour être prises en compte lors de l'élaboration d'annotations débarquées : la contextualité est limitée à un jeu d'interactions défini par des choix initiaux. Le choix du modèle est justifié chez ces auteurs par des hypothèses sur la nature compositionnelle des niveaux, et sur leur autonomie réelle, et non méthodologique.

de hiérarchies conflictuelles reste limité, et si les conflits d'annotation peuvent être résolus en distinguant différents sous-ensembles eux-mêmes hiérarchiques[29].

L'annotation en nœuds-bornes fait le choix inverse de conserver l'intégration, dans une structure commune, des annotations, au prix d'une perte de l'expression directe de la dominance. C'est l'axe de la précédence qui est privilégié. Toutes les annotations sont exprimées sur un axe linéaire commun, et peuvent donc être rapportées les unes aux autres, c'est-à-dire contextualisées.

Si la structure de données reste un arbre, la modélisation impliquée par l'utilisation de nœuds-bornes n'est dès lors plus un arbre, mais un graphe[30]. En effet, les arcs pleins d'une part et pointillés d'autre part de la figure 4 ci-dessus sont deux types d'arcs différents (resp. « normaux » et « nœuds-bornes ») du fait des limitations d'XML, mais du point de vue du corpus construit, ce sont deux unités, au même titre. On peut dès lors représenter la structure de données induite par l'utilisation de nœuds-bornes comme un graphe (figure 5) où la distinction entre les deux types de nœuds est annulée :

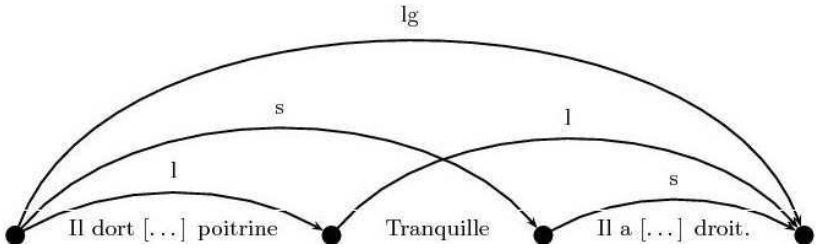

Figure 5 : une représentation commune de l'annotation hiérarchique et de l'annotation en nœuds borne de la figure 4 par un graphe d'annotation.

29. Certains conflits peuvent être dus à des unités isolées, non incluses dans une hiérarchie. En pratique, l'annotation débarquée est principalement utilisée dans des dispositifs où l'on confie à chaque arborescence un niveau de description : on reproduit et réifie alors la distinction des niveaux, plutôt que de la remettre en cause.

30. Voir Loiseau (2007).

Sur la figure 5, la segmentation est représentée uniquement sur l'axe de la précédence – le seul que les deux types d'annotation ont en commun. Un seul nœud sépare deux segments de textes, même si plusieurs segmentations y sont rattachées, puisqu'il n'y a qu'une seule coordonnée entre deux nœuds textes sur cet axe.

Cette structure est très proche des graphes d'annotation (AG) proposés par Bird *et al.* (2001). Les auteurs pointent des besoins proches de ceux qui ont motivé CorpusReader. Mettant l'accent sur les caractéristiques formelles des modèles d'annotation, ils montrent la nécessité, pour disposer d'un modèle d'annotation expressif, de privilégier la linéarité sur la compositionnalité, de permettre la redondance (par exemple, lorsque plusieurs analyseurs sont utilisés conjointement, une partie de l'information est redondante) et l'annotation partielle (c'est-à-dire des zones moins densément annotées ou inégalement couvertes). Ils reprochent aux modélisations basées sur un arbre de privilégier la dominance :

> Les modèles arborescents traitent tous la relation de dominance comme fondamentale. Nous pensons que cela conduit à des difficultés non triviales (Bird & Liberman, 2001).

Les graphes d'annotation sont des graphes orientés[31], non cycliques[32], étiquetés[33] et partiellement ou totalement ancrés[34] :

Les graphes d'annotation sont donc un formalisme plus adapté à la représentation d'une pluralité d'interprétations, et

31. Un graphe orienté est un graphe où les arêtes distinguent un nœud d'origine et un nœud de destination. Ici, l'orientation des arêtes est fondée sur la distribution des nœuds sur un axe chronologique : le nœud d'origine d'une arête est antérieur chronologiquement à son nœud de destination. Dans les graphes d'annotation, cet axe chronologique est fondé sur le son ou la vidéo : on peut tout autant le fonder sur la linéarité du signifiant graphique. Les graphes d'annotation ont été construits pour la représentation de corpus oraux, mais leur formalisme est emprunté aux *chart parsers*.

32. Un graphe non cyclique est un graphe où aucun chemin, le long des arêtes, ne permet de revenir sur ses pas.

33. Les arêtes reçoivent des étiquettes.

34. Les nœuds reçoivent une estampille temporelle qui les indexe dans la linéarité du signifiant.

donc d'une contextualité étendue, que le modèle arborescent. Cependant aucun vocabulaire standard, remplissant le rôle de la TEI dans le cadre d'XML, n'est proposé dans le cadre du formalisme des graphes d'annotation. Les outils existants sont en nombre limité, sans commune mesure avec l'expérience et l'outillage disponible dans le contexte d'XML. Le dispositif adopté avec CorpusReader consiste à opérer un compromis : un graphe d'annotation est représenté dans la linéarité d'un document XML.

Cette solution présente néanmoins plusieurs inconvénients. En premier lieu la souplesse introduite implique une moindre capacité de contrôle de la structure (par les DTD ou schémas). L'abandon de l'arborescence comme moyen de modélisation rend moins aisée la manipulation du document avec les outils de traitement standard pour XML. Le gain en expression de la contextualité et donc en richesse empirique de l'objet représenté entraîne un traitement plus coûteux du document. C'est pourquoi CorpusReader propose plusieurs mécanismes pour utiliser les nœuds-bornes et les transformer en annotation normale dès que le choix d'une contextualité (définie et limitée par une analyse donnée) le rend possible (voir ci-dessous).

6. Contextualiser : fusionner les annotations

Le premier objectif de CorpusReader est de permettre l'intégration des annotations produites par des analyseurs tiers dans un corpus déjà annoté.

Un exemple permettra d'illustrer la stratégie adoptée : l'ajout d'une annotation morphosyntaxique, réalisée par Cordial, aux deux derniers vers du *Dormeur du val*, annoté avec la TEI.

La tâche d'intégration de l'annotation tierce se décompose en trois étapes (résumées dans les figures 6 à 8 ci-dessous).

Dans un premier temps, une procédure d'extraction du texte à annoter est nécessaire. Cette extraction peut être réalisée à faible coût (par exemple, avec XSLT). Elle permet de prendre en compte les exigences de chaque analyseur (dans notre exemple, la suppression des retours à la ligne en début de vers,

que Cordial interprète comme des changements de phrases). Elle permet également de soustraire à l'analyse certaines portions du document, comme l'en-tête.

Dans un second temps l'annotation produite est convertie en XML en conservant le texte d'origine restitué par l'analyseur (la « forme fléchie » dans les sorties de Cordial). Cette conversion est là encore réalisable à faible coût, souvent au moyen de simples expressions régulières, comme dans cet exemple.

Une fois cette conversion réalisée, CorpusReader peut réaliser la fusion entre les deux annotations. Pour cela, il faut d'une part aligner le texte commun des deux documents XML caractère par caractère[35] puis insérer les informations nouvelles fournies par l'instrument dans le corpus de destination. En cas de conflit entre hiérarchies, CorpusReader convertit la nouvelle annotation en nœuds-bornes.

```
<text>
  <lg>
    <l>Il dort dans le soleil, la main sur sa poitrine</l>
    <l>Tranquille. Il a deux trous rouges au côté droit.</l>
  </lg>
</text>
```

```
Il dort dans le soleil, la main sur sa poitrine tranquille. Il a deux trous rouges au côté droit.
```

Figure 6 : étape 1 : extraction du texte à annoter.

35. Un algorithme de distance d'édition permet de pallier les altérations fréquemment introduites dans le texte original « restitué » en sortie de l'analyseur.

```
        ==== DEBUT DE PHRASE ====
Il      il      PPER3S
dort    dormir  VINDP3S
dans    dans    PREP
le      le      DETDMS
soleil  soleil  NCMS
        ==== FIN DE PHRASE ====
```

```
<s>
<w lemma="il" ana="PPER3S">Il<w>
<w lemma="dormir" ana="VINDP3S">dort<w>
<w lemma="dans" ana="PREP">dans<w>
<w lemma="le ana="DETDMS">le<w>
<w lemma="soleil" ana="NCMS">soleil<w>
</s>
```

Figure 7 : étape 2 : transcodage ligne à ligne des sorties tabulées de Cordial en XML.

```
<l><s type="start"></s><w lemma="il" ana="PPER3S">Il<w>
<w lemma="dormir" ana="VINDP3S">dort<w> [...]</l>
<l> [...] <s type="end"></s> [...]</l>
```

Figure 8 : étape 3 : fusion des annotation. Les éléments conflictuels de l'annotation tierce sont convertis en nœuds-bornes (cf. l'élément « s »).

L'un des avantages de cette méthode est que seule la seconde étape est spécifique à un format idiomatique ; de nouveaux instruments peuvent être associés à très faible coût. Cette stratégie d'intégration d'annotation repose sur l'utilisation d'un format commun. On peut la distinguer d'une autre méthode consistant à redéfinir les instruments d'annotation, ou du moins à les intégrer dans un dispositif logiciel, pour permettre la multi-annotation. L'« intégration par le format » adoptée ici est également proposée dans la plate forme Atlas (Bird *et al.*, 2000), dans le cadre des graphes d'annotations. À l'inverse,

l'« intégration par les instruments » est le choix que fait GATE : cette plate-forme définit des « interfaces », c'est-à-dire des comportements-types des logiciels, que des analyseurs doivent respecter de façon à interagir. La multi-annotation ne peut être réalisée qu'avec des analyseurs partiellement (ré)écrits pour ce cadre. Le format d'annotation quant à lui peut alors être interne et spécifique à la plate-forme. CorpusReader fait le choix inverse : c'est le format qui est contraint, pas le comportement des logiciels d'annotation.

L'enjeu de cette alternative est la réutilisabilité, puisque les plates-formes aux formats d'annotation idiomatiques ne peuvent pas inter-opérer et reproduisent, au niveau de la multi-annotation, la babélisation des formats. De plus, le choix d'une intégration « par les instruments » impose un travail de réfection logicielle des instruments existants[36], tandis que l'intégration « par les formats » rend plus immédiatement disponible, à faible coût, l'ensemble de ces instruments. Dans le cas de l'intégration « par les instruments », la multi-annotation est vue comme un nouveau stade logiciel impliquant une redéfinition d'instruments : l'instrument reste premier.

L'intégration réalisée ici entre annotations est une intégration « faible » au sens où les divergences de segmentation en paliers ont été acceptées et que l'on n'a pas cherché à unifier les jeux d'étiquettes d'instruments concurrents sur un même niveau. Cette unification des analyses est l'objectif de projets comme Amalgam (Atwell *et al.*, 1994), dans une perspective notamment d'évaluation des analyseurs. Ici, c'est au contraire d'une certaine façon la diversité des analyses qui est supposée intéressante, et la possibilité d'utiliser chacune pour ce en quoi elle est performante ou adaptée à une hypothèse.

36. Même si les analyseurs sont seulement modifiés dans leur interface, pas dans le cœur de qu'ils calculent, imposer la contrainte d'un accord logiciel est coûteux : certains instruments peuvent mal s'y prêter ; si les codes sources ne sont pas publics, cette réfection est impossible ; bref toute condition de ce type réduit d'emblée la diversité des instruments utilisables. On réduirait également cette diversité en attendant de l'instrument qu'il gère un format imposé, comme XML.

7. Construire les observables

Outre la constitution d'une multi-annotation par fusion d'analyses tierces, le second rôle de CorpusReader est la manipulation et l'interrogation de l'annotation constituée : sélection de sous-ensembles de l'annotation, de sous-corpus ; quantification de phénomènes ; projection dans des structures de données (listes de fréquences, matrices, graphes) utiles pour des analyses quantitatives.

7.1. Choix d'une API

XML peut être manipulé par deux principales méthodologies informatiques (dites « API », pour *application programming interface*, interface de programmation d'application), qui proposent chacune une représentation logique du document (une sémantique de traitement).

Dans la première de ces méthodologies, DOM (*Document Object Model*), le document est représenté comme un arbre, que l'on parcourt principalement en utilisant les relations de dominance. Dans la seconde, SAX (*Simple API for XML*), le document est représenté comme une « liste » de briques correspondant aux différents constituants du document (un élément est représenté par deux briques : une balise ouvrante et une balise fermante, correspondant au « tag ouvrant » et au « tag fermant » de la forme sérialisée) ; l'ordre des briques dans cette liste représente l'ordre d'une traversée profondeur-droite de l'arbre (cet ordre est représenté par une flèche dans la figure 9). C'est également l'ordre dans lequel apparaissent les briques dans un fichier sérialisant le document. C'est la relation de précédence, cette fois, qui est donc privilégiée, et les relations hiérarchiques ne sont pas représentées directement. Elles peuvent cependant être reconstituées à partir de la liste de briques reçue. Les deux méthodologies mettent donc chacune l'accent sur une contextualité différente, et l'instrumentalisent : hiérarchique (inclusions) pour DOM, linéaire (successions) pour SAX.

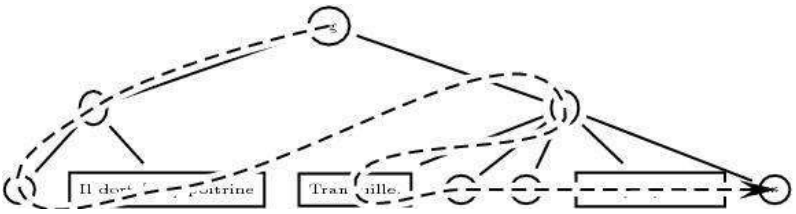

Figure 9 : l'ordre de la traversée de l'arbre dans l'API SAX.

Cette seconde méthodologie offre plusieurs avantages. En premier lieu, un traitement en flux grâce à l'API SAX permet de ne pas limiter la taille des corpus traités, puisque la liste peut être alimentée « brique par brique », sans charger le corpus entier en mémoire. Seules les informations concernées par la requête sont extraites et mémorisées.

De plus, l'API SAX est particulièrement adaptée à l'utilisation de nœuds-bornes. En effet, si le recours aux nœuds-bornes donne davantage d'expressivité, il rend moins aisé la manipulation de l'arbre avec les langages standard d'extraction ou de transformation comme XPath et XSLT. En effet, ces langages reposent sur l'exploitation de la dominance pour désigner le contenu d'une unité, ou exprimer une itération. De plus, seul un sous-ensemble restreint de l'annotation est utilisé ainsi dans une requête donnée – particulièrement dans le cadre d'une tâche de quantification. Il est donc utile, pour bénéficier des langages et outils existants de transformer préalablement certains nœuds-bornes en éléments hiérarchiques.

L'identification et la conversion d'une paire de nœuds-bornes en éléments hiérarchiques ou « normaux » est particulièrement facile si on accède au contenu sur l'axe de la précédence : le nœud-borne représentant un début d'unité est transformé en élément ouvrant, et le nœud-borne représentant une fin d'unité en élément fermant. L'opération est strictement homologue à l'identification de paire élément ouvrant/élément fermant. Le cadre d'un traitement séquentiel rend également aisée la transformation en nœuds-bornes des éléments standard qui entrent en conflit avec ce nouvel élément.

Ainsi on choisit, en fonction des besoins de traitement, quel sous-ensemble de l'annotation doit être transformé en

éléments standard. Autrement dit, l'arbre XML peut être composé à la volée et au cas par cas, à partir des informations présentes dans les nœuds-bornes, pour être adapté à une requête précise : le choix de la composition de l'arbre XML ne correspond plus à une modélisation mais à une tâche.

7.2. Exprimer une requête

L'expression de requêtes est réalisée en exploitant les propriétés de la méthodologie SAX. Les fonctions de CorpusReader sont implémentées dans des filtres ; ceux-ci peuvent être cumulés pour composer un enchaînement de traitements, dit pipeline, qui reçoit le flux de briques SAX[37]. Ces filtres peuvent réaliser différentes opérations : modifier le flux, en extraire des informations, changer la structure du document. La coopération des filtres dans un pipeline permet de définir des traitements complexes.

On peut répartir ces filtres en trois ensembles qui correspondent à des utilisations types de CorpusReader.

En premier lieu, certains filtres permettent d'utiliser des langages standard et généralistes, comme XSLT ou XQuery – qui permettent de transformer la structure du document, d'en extraire des sous-corpus ou des quantifications – ou de valider le document avec des Schémas[38]. De ce point de vue, CorpusReader peut-être utilisé simplement comme un cadre pour mettre en œuvre ces langages.

La plupart des filtres cependant ont une fonction précise, qui leur permet de se combiner facilement en pipeline complexe : décompter des phénomènes, extraire des sous-corpus, manipuler la structure de l'arbre, convertir les structures de données dans plusieurs formats. Ces filtres opèrent au niveau du flux de briques et sont donc plus performants. Ils sont spécialisés : un seul filtre par exemple réalise les tâches de quantification. Le décompte de phénomènes dans le corpus peut s'exprimer de différentes façons, incluant l'utilisation du

37. Une documentation complète des filtres est disponible à l'adresse <http://panini.u-paris10.fr/~sloiseau/CR/filtres-intro.html>.

38. Les filtres de ce type reposent sur des composants externes, comme Saxon de M. Kay ou Jing de J. Clark.

langage XPath : tout phénomène exprimé dans l'infoset XML[39] (occurrence d'élément, d'attribut, ou de « patron » plus complexe dans l'arbre) peut donc être quantifié.

Un filtre enfin prend comme argument du code Java et l'exécute à la volée sur les briques reçues. CorpusReader peut donc être utilisé pour tester un prototype de filtre ou pour utiliser l'API SAX sans avoir à gérer les détails de sa mise en œuvre. CorpusReader peut importer dans un pipeline tout filtre SAX extérieur : la collaboration avec d'autres outils est ainsi réalisée au niveau de l'API elle-même.

Deux filtres sont particulièrement importants : l'un est spécialisé dans la transformation des nœuds-bornes en annotation hiérarchique, et peut être utilisé, en amont d'un autre filtre, pour lui permettre de s'appuyer sur une arborescence ; l'autre permet d'intégrer dans le corpus, selon la procédure présentée ci-dessus, les annotations tierces contenues dans un autre document.

7.3. Distinguer deux états du corpus : corpus « accumulateur » et corpus d'exploration

L'élément central du dispositif est la distinction de deux corpus : le corpus accumulateur ou « contextualisant », obtenu par fusions successives d'analyses tierces, et des corpus d'exploration, correspondant à des sous-ensembles du premier en fonction des expériences et des hypothèses[40]. La distinction de ces deux états permet de ne pas faire peser de contraintes de maniabilité sur le corpus accumulateur, qui ne doit répondre qu'à des contraintes d'expressivité ou de contextualité. Elle permet, en retour, de ne pas faire peser de contraintes d'expressivité sur les états d'exploration. CorpusReader peut finalement être défini comme un outil permettant d'une part la

39. Le modèle de données d'XML (Cowan & Tobin, 2004).

40. Cette distinction est différente de la distinction entre corpus de travail et corpus de référence (Rastier, 2001) : il ne s'agit pas de la relation entre un ensemble de contraste et un ensemble décrit, mais de la relation entre un objet empirique « complet » mais insaisissable et un objet empirique observable, selon une hypothèse et au prix d'une réduction de sa richesse.

constitution du corpus accumulateur, d'autre part le passage du corpus accumulateur, où la contextualité est maximale, aux corpus d'analyse, qui correspondent à des contextes choisis, construits parmi les annotations disponibles. Ainsi il permet de découpler les exigences d'expressivité et de maniabilité.

Le contexte est alors défini non seulement par sa longueur, c'est-à-dire par la sélection d'un empan, mais aussi par son « épaisseur », c'est-à-dire par la sélection d'un sous-ensemble d'annotations parmi l'ensemble des annotations disponibles. Le corpus d'exploration représente donc une sélection dans l'épaisseur du contexte.

7.4. Donner accès à toutes les propriétés de l'objet : transformer une API en outil

Une autre propriété de ce dispositif est de chercher à faire d'une API un outil. Dans la mesure où XML est utilisé comme format d'annotation, toutes ses propriétés doivent être accessibles puisqu'elles intéressent l'objet constitué. Or une API est par définition le type de composant logiciel où sont accessibles toutes les propriétés de la structure de données. Ainsi CorpusReader est construit sur l'API SAX, dont il cherche à exprimer plutôt qu'à abstraire la logique. Dans une certaine mesure CorpusReader est donc un dispositif visant à rendre simple l'utilisation de cette API.

En tant que dispositif expérimental, CorpusReader propose une typologie d'« objets » représentant la manipulation du corpus. Il est intéressant de noter que les architectures permettant une multi-annotation proposent de nouveaux types d'objets, qui ne modélisent ni des concepts linguistiques, ni des concepts informatiques, mais correspondent davantage à une modélisation du processus d'interprétation. Par exemple, GATE est construit autour d'une distinction entre les « ressources » (LR, *Language Ressources*) et les « composants logiciels » (LE, *Language Engeeniring*). LinguaStream propose les concepts de « vue » et de « modèle d'analyse ». Ces distinctions peuvent être de réelles propositions théoriques ou méthodologiques. La distinction entre LE et LR peut sembler de bon sens mais c'est un choix théorique sur les éléments manipulés par le linguiste :

elle réifie une distinction entre données et programme issue de l'Intelligence Artificielle.

De ce point de vue, une caractéristique de CorpusReader est de proposer aussi peu d'abstraction et donc d'objets que possible[41] : les objets proposés reflètent des concepts de bas niveau issus de l'API, comme la notion de filtre SAX[42].

5. Conclusion

Trois choix principaux caractérisent CorpusReader.

Le premier est de baser la constitution de la multi-annotation sur un alignement des sorties et non sur une intégration des instruments d'annotation. Ce choix permet de réutiliser l'ensemble des instruments d'annotation existants. Une contextualité est produite en partageant entre de multiples instruments la responsabilité de l'annotation.

Un second choix est de ne pas limiter par avance les possibilités de croisements et de produire une contextualité aussi forte que possible. La contextualité est une propriété centrale de l'artefact produit et non pas une propriété de second rang, obtenue grâce à un dispositif logiciel, comme dans le cas de l'annotation débarquée.

Un troisième choix est de ne pas abstraire le codage dans des représentations de haut niveau de la manipulation du corpus, mais au contraire de rendre accessible ce codage. Le format du codage est une des propriétés de l'objet observable. L'objectif est de donner un statut d'objet à XML comme structure formelle et donner un statut d'outil à une API, en

41. Les objets introduits par ce dispositif portent sur des états différents du corpus (l'opposition entre corpus d'accumulation et corpus d'exploration) et non sur l'activité interprétative ou les composants de l'objet.

42. LinguaStream propose également une représentation d'un traitement par une succession de « filtres », semblable à ceux de CorpusReader, cependant il s'agit d'un objet plus abstrait, ne correspondant pas à une unité de traitement de l'API SAX : les filtres de LinguaStream admettent par exemple plusieurs flux en entrée et en sortie.

cohérence avec l'hypothèse que l'annotation est elle-même une interprétation[43].

Ces choix entraînent la distinction entre deux états du corpus, un état contextualisant et un état d'exploration, qui permet de découpler les exigences d'expressivité et de maniabilité. L'arborescence XML peut-être composée à la volée : elle correspond aux besoins d'une tâche, et non à une modélisation de l'objet. Une « requête » CorpusReader est dès lors un passage du premier état dans le second.

Du point de vue des propriétés empiriques des artefacts qu'il produit, le contexte reçoit dans ce dispositif une détermination particulière : il désigne une pluralité d'interprétations coexistantes. Le contexte produit par CorpusReader est issu de catégories descriptives (les annotations de différents instruments mis à contribution). En quelque sorte, ce dispositif permet de faire d'un aboutissement (les catégories descriptives réifiées par un instrument) un point de départ ou une nouvelle « matière première » pour la description. Ce « recyclage » des ressources traduit, somme toute, une propriété forte de la description en sciences humaines, où l'on n'accède jamais à une matière qui ne soit pas déjà informée d'interprétations.

Enfin, l'enjeu est peut-être la capacité à décrire un texte annoté non seulement en tant qu'il est un artefact permettant d'observer un texte « préexistant », mais également en tant qu'il est lui-même un nouvel objet linguistique, dans la mesure où l'annotation crée une nouvelle matérialité participant à la textualité. S'il est vrai qu'il n'y a pas de langue neutre, indépendante de sa matérialité, alors l'annotation est une

43. On peut opposer ce choix de reposer fortement sur un format donné à l'architecture proposée par GATE (Voir (Bontcheva *et al.*, 2004), p. 353 et p. 357 : « GATE users are isolated from the ways in which LRs are stored ») : dans ce dispositif expérimental, différents formats de stockage sont interchangeables, notamment les bases de données ou XML. Ceci implique une abstraction du format dans la représentation donnée du corpus à l'utilisateur. Le choix de CorpusReader est au contraire de considérer que l'ensemble des propriétés du format intéresse le corpus comme artefact linguistique et qu'il ne doit pas être soustrait à l'analyse. Pour autant, comme nous l'avons vu, le formalisme du format n'est pas considéré comme un modèle théorique dans le dispositif adopté, mais bien comme un format de représentation.

propriété de l'objet et pas seulement un outil d'analyse et la prise en compte de cela importe sans doute, au moins à titre d'horizon, à la description de corpus annotés.

Bibliographie

Barnard D., Burnard L., Gaspart J.-P., Price L., Sperberg-McQueen C., Varile N. (1992). *Notes on SGML Solutions to Markup Problems*, Rapport technique.

Biber D. (1995). *Dimensions of register variation : a cross-linguistic comparison*. Cambridge : Cambridge University Press.

Bilhaut F. & Widlöcher A. (2006). « LinguaStream : An Integrated Environment for Computational Linguistics Experimentation », *Proceedings of the 11th Conference of the European Chapter of the ACL*, pp. 95-98.

Bird S., Day D., Garofolo J., Henderson J., Laprun C., Liberman M. (2000). « ATLAS : A Flexible and Extensible Architecture for Linguistic Annotation », *Proceedings of the Second International Conference on Language Resources and Evaluation*, pp. 1699-1706.

Bird S. & Liberman M. (2001). « A formal framework for linguistic annotation », *Speech Communication* 33-1/2 : 23–60.

Bontcheva K., Tablan V., Maynard D., Cunningham H. (2004). « Evolving GATE to meet new challenges in language engineering », *Natural language Enginering* 10-3/4 : 349-373.

Bray T., Paoli J., Sperberg-McQueen C.-M., Maler E. (2000). *Extensible Markup Language (XML) - 1.0, second edition, W3C Recommendations*.

Burnard L. (1995). « Text Encoding for Information Interchange – An Introduction to the Text Encoding Initiative », *Proceedings of the Second Language Engineering Conference*.

Canguilhem G. (2003 [1965]). *La connaissance de la vie*. Paris : Vrin.

Carletta J., Kilgour J., O'Donnel T., Evert S., Voormann H. (2003). « The NITE Object Model Library for Handling Structured Linguistic Annotation on Multimodal Data Sets », *Proceedings of the EACL Workshop on Language Technology and the Semantic Web*.

Coseriu E. (2001). *L'homme et son langage*. Leuven : Peeters.

Cowan J. & Tobin R. (2004). *XML Information Set (Second Edition)*, W3C Recommandation.

DeRose S. (2004). « Markup Overlap : A Review and a Horse », *Proceedings of Extreme Markup Language*.

Gadet F. (2003). *La variation sociale en français*. Paris : Ophrys.

Glessgen M.-D. (2007) *Linguistique romane*. Paris : Armand Colin

Guiraud P. (1969). *Essai de stylistique.* Paris : Klincksieck.

Habert B. (2005a). « Portrait de linguiste(s) à l'instrument », revue électronique *Texto !*, X-4.

Habert B. (2005b). *Instruments et ressources électroniques pour le français*. Paris : Ophrys.

Habert B. & Zweigenbaum (2002). « Régler les règles », *TAL*, 43-3 : 83-105.

Heiden S. (2006). « Un modèle de données pour la textométrie : contribution à une interopérabilité entre outils », *Actes des 8èmes Journées internationales d'Analyse Statistique des Données Textuelles*. Besançon : Presses Universitaires de Franche-Comté, vol. 1, pp. 487-498.

Lara L.-F. (1983). « Le concept de norme dans la théorie d'Eugénio Coseriu », ***in*** E. Bédard & J. Maurais (éds) *La norme linguistique*. Québec : gouvernement du Québec.

Loiseau S. (2006). *Sémantique du discours philosophique chez Deleuze : du corpus aux normes*, Thèse de doctorat, Nanterre : Université Paris X Nanterre.

Loiseau S. (2007). « A formalism for representing together milestone and normal annotation », *Proceedings of Extreme markup languages*.

Loiseau S., Poudat C., Ablali D. (2006). « Exploration contrastive de trois corpus de sciences humaines », *Actes des 8èmes Journées d'analyse des données textuelles,* Besançon : Presses Universitaires de Franche-Comté, vol. 2, pp. 631–643.

Poudat C. (2006). *Étude contrastive de l'article scientifique de revue linguistique dans une perspective d'analyse des genres,* Thèse de doctorat, Orléans : Université d'Orléans.

Rastier F. (2001). *Arts et sciences du texte*. Paris : Presses Universitaires de France.

Rastier F. (2004). « Doxa et lexique en corpus – Pour une sémantique des « idéologies » », *Actes des Journées scientifiques en linguistique 2002-2003 du CIRLEP*. Reims : Presses Universitaires de Reims.

Rastier F. (2005). « Enjeux épistémologiques de la linguistique de corpus » *in* Williams G. (éd.) *La linguistique de corpus*. Rennes : Presses Universitaires de Rennes, pp. 31-47.

Sperberg-McQueen C. M. (2005). « XML and Semi-Structured Data », *ACM Queue*, 3-8.

Sperberg-McQueen C.-M. & Burnard L. (2001). *TEI P4, Guidelines for Electronic Text Encoding and Interchange.* Virginia : University of Virginia Press.

Teich E., Hansen S., Fankhauser P. (2001). « Representing and querying multi-layer annotated corpora », *Proceedings of IRCS Workshop on Linguistic Databases*. Pennsylvania : University of Pennsylvania.